

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
КАФЕДРА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ЭНЕРГЕТИЧЕСКИХ СИСТЕМ

**Мугайских Александр Всеволодович**

Магистерская диссертация

**Задачи маршрутизации транспорта на сети мегаполиса**

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа «Математическое и информационное обеспечение экономической деятельности»

Заведующий кафедрой,  
доктор физ.-мат. наук,  
профессор

Захаров В. В.

Научный руководитель,  
доктор физ.-мат. наук,  
профессор

Захаров В. В.

Рецензент,  
начальник Отдела систем анализа и прогнозирования  
Дирекции региональных продаж,  
ПАО «Газпром нефть»

Черницын И. Г.

Санкт-Петербург

2017

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Задача маршрутизации транспорта при переменных значениях времени перехода на сети</b>	<b>7</b>
1.1 Обзор литературы . . . . .	7
1.2 Постановка задачи TD-MDVRP . . . . .	13
1.3 Эвристические алгоритмы решения задачи MDVRP . . . . .	17
1.4 Генетический алгоритм решения задачи MDVRP . . . . .	20
1.5 Динамическая адаптация генетического алгоритма . . . . .	28
<b>2 Задача распределения транспортных потоков</b>	<b>32</b>
2.1 Постановка задачи распределения транспортных потоков . . . . .	32
2.2 Решение задачи распределения транспортных потоков на улично- дорожной сети г. Санкт-Петербурга . . . . .	35
<b>3 Эксперимент</b>	<b>38</b>
3.1 Задача маршрутизации транспорта при переменных значениях потоков между районами отправления и прибытия . . . . .	38
3.2 Задача маршрутизации транспорта при переменных значениях времени перехода между узлами сети . . . . .	42
<b>4 Заключение</b>	<b>49</b>
<b>Литература</b>	<b>51</b>
<b>Приложение</b>	<b>59</b>

## Введение

В настоящее время компании всё чаще сталкиваются с задачами разработки прикладных моделей и методов управления цепями поставок, а так же оптимизации уже существующего аналитического аппарата. Значительную роль в решении этих задач может играть прикладная математическая (интеллектуальная) логистика. В общем случае, логистика является наукой о планировании, организации, управлении и контроле движения материальных и информационных потоков в пространстве и во времени от их первичного источника до конечного потребителя. Особое внимание исследователи уделяют транспортной логистике как инструменту, который позволяет при том же объёме используемых ресурсов увеличить прибыль. Под транспортной логистикой понимают совокупность процессов погрузки-разгрузки, экспедирования и логистических операций. Транспортная логистика рассматривает задачи обеспечения технической и технологической сопряжённости участников транспортного процесса, совместного планирования производственного, транспортного и складского процессов, а так же задачи определения рациональных маршрутов доставки [1]. Центральное место в области транспортной логистики занимает задача маршрутизации транспортных средств. Целью данной задачи является построение множества оптимальных маршрутов для набора транспортных средств, расположенных в депо (на складе) и обслуживающих заданное множество клиентов. Выбор неудачного маршрута при планировании влечёт за собой дополнительные накладные затраты.

Данная работа посвящена задаче маршрутизации транспортных средств на сети мегаполиса. Разработанные к настоящему времени эвристические и метаэвристические алгоритмы строят маршруты обхода клиентов, не учиты-

вая текущую информацию о трафике. В большинстве математических моделей предполагается, что матрица временных затрат на перемещение между узлами сети (матрица переходов) является постоянной, то есть время перехода между каждой парой узлов неизменно в течение всего периода маршрутизации. Некоторые модели допускают зависимость времени движения на сети от периода дня, однако выполнение данного свойства достигается приближенными методами: например, умножением матрицы переходов на коэффициент, соответствующий периоду дня. Как результат, данная аппроксимация слабо отражает реальную загруженность сети и может привести к решению задачи маршрутизации, которое не является оптимальным, а в некоторых постановках задач – к неявляющемуся допустимым (в случае ограничений типа «временные окна» и других).

В качестве примера задачи транспортной маршрутизации в данной работе рассмотрена задача MDVRP (Multiple Depot Vehicle Routing Problem), также известная как задача маршрутизации с несколькими депо. Все задачи этого класса сводятся к задачам комбинаторной оптимизации и являются NP-трудными. В тестовых примерах большой размерности задача становится трансвычислительной, поэтому точные методы не применимы для её решения. По этой причине применяют эвристические и метаэвристические алгоритмы. Время работы данных методов существенно меньше, чем у точных алгоритмов, а их решения уступают в экономичности оптимальному, как правило, несколько процентов. В целях сравнения эффективности эвристических алгоритмов исследователи проводят эксперименты на тестовых задачах из стандартных библиотек.

В первой главе представлена математическая модель и приведена клас-

сификация известных методов решения динамической задачи TD-MDVRP (Time Dependent Multiple Depot Vehicle Routing Problem) с изменяющейся во времени матрицей переходов. Также в этой главе подробно описано применение генетического алгоритма (ГА) для построения маршрутов и исследован экспериментальный уровень временной состоятельности генетического алгоритма. На основе данного критерия разработана процедура динамической адаптации генетической эвристики решения задач транспортной маршрутизации. В главе 2 предложена объединённая постановка задач распределения транспортных потоков TAP (Traffic Assignment Problem) и маршрутизации транспорта при переменных значениях потоков между районами отправления и прибытия в течение всего периода маршрутизации. Результаты, полученные в главе 2, опираются на методы вычисления распределения транспортных потоков при использовании BPR-функции задержки, описанные в [2, 3]. Третья глава посвящена экспериментам, проведённым на улично-дорожной сети г. Санкт-Петербурга. Были сгенерированы и решены тестовые задачи MDVRP при переменных значениях времени перехода между узлами сети.

Основными результатами выпускной квалификационной работы магистра являются:

1. Описана динамическая задача маршрутизации и исследованы динамические модели маршрутизации транспортных средств на сети мегаполиса.
2. Разработана процедура динамической адаптации генетического алгоритма для задач данного класса.
3. Проведена экспериментальная оценка уровня временной состоятельности генетического алгоритма.
4. Исследована и представлена в виде графа улично-дорожная сеть г. Санкт-

Петербурга, сгенерированы тестовые задачи TAP и MDVRP на сети города и получены их численные решения.

5. Решена задача распределения транспортных потоков при переменных значениях потоков между районами отправления и прибытия г. Санкт-Петербурга.
6. Решена задача маршрутизации транспорта при переменных матрицах переходов на сети г. Санкт-Петербурга.
7. Разработан программный модуль для численного решения задач класса TD-MDVRP.
8. Получено экспериментальное подтверждение эффективности предложенной модели в задачах маршрутизации транспорта на сети мегаполиса.

# 1 Задача маршрутизации транспорта при переменных значениях времени перехода на сети

## 1.1 Обзор литературы

Задача маршрутизации транспорта при переменных значениях времени перехода на сети (TDVRP) – одна из модификаций классической задачи маршрутизации транспорта (VRP). Фактор трафика в условиях загруженной улично-дорожной сети мегаполиса играет существенную роль и не должен быть проигнорирован в рамках построения маршрутных планов. Задача маршрутизации транспорта (VRP) состоит в нахождении маршрутных планов для множества идентичных машин, расположенных в депо и обслуживающих заданное множество клиентов так, чтобы каждый клиент был посещён ровно один раз и суммарные затраты на маршруты движения всех машин были минимальны. Под затратами на движение далее будем понимать время машин в пути. С момента формулировки задачи VRP в 1959 появилось большое число разновидностей этой задачи, интерес к которой вызван, в первую очередь, ее практической значимостью. На данный момент выделяют следующие виды задачи VRP, имеющие различия в постановках:

- Open VRP (задача VRP без необходимости возвращаться в депо) [4]
- Multiple Depot VRP (задача VRP с несколькими депо) [5]
- VRP with Pick-Up and Delivering (задача VRP с вывозом и доставкой) [6]
- Capacitated VRP (задача VRP с учётом ограничения на вместительность машин) [7]
- VRP with Time Windows (задача VRP с временными окнами) [7]

- VRP with Backhauls (аналогична VRP with Pick-Up and Delivering, но с идентичным видом товаров у всех клиентов) [7]
- Periodic VRP (задача VRP с несколькими периодами) [8]
- Stochastic VRP (задача VRP со случайными параметрами: например, спросом клиентов в товаре) [9]
- Split Delivery VRP (задача VRP с возможностью обслуживания одного клиента несколькими транспортными средствами) [10]
- VRP with Satellite Facilities (задача VRP с возможностью дозагрузки автомобиля на маршруте) [11]
- Dynamic VRP (задача VRP с динамически появляющимися заказами) [12]

В модификации Time Dependent VRP скорость движения по любому ребру сети не является постоянной в течение всего периода маршрутизации, т.е. затрачиваемое на переход между двумя узлами сети время является функцией, зависящей не только от расстояния между узлами.

Задаче маршрутизации транспорта при переменных значениях времени перехода на сети (TD-VRP) уделяется мало внимания со стороны исследователей, несмотря на то, что модель TD-VRP позволяет учитывать загруженность улично-дорожной сети в отличие от классической модели MDVRP. Впервые математическая формулировка задачи TD-VRP в рамках смешанного целочисленного линейного программирования была предложена в работе [13]. Авторы работали одновременно над задачами TD-VRP и TD-TSP (Time Dependent Travelling Salesman Problem), в которой мощность множества доступных машин равна одному. Модель также позволяла учитывать следующие виды ограничений, такие как: временные окна у клиентов, грузо-



подъёмность машин и возможность ожидания у клиентов. В работе случайно сгенерированные тестовые задачи малого размера (до 25 вершин) были решены эвристикой ближайшего соседа и методом ветвей и отсечений. Функция времени перехода по ребру сети представляла ступенчатую функцию из 2 периодов и была постоянна на протяжении каждого из периодов. В статье [14] авторы разработали алгоритм динамического программирования с элементами эвристики для решения задачи TD-VRP. Описанный подход подходил для разных видов функций времени перехода по ребру, однако эксперименты проводились только для ступенчатых функций как и в работе [13]. Другая постановка задачи TD-VRP была сформулирована исследователями в статье [15]: узлу  $k$  на сети соответствовала постоянная скорость движения  $r_{kT}$  в период  $T$ , а время перехода из узла  $i$  до  $j$  вычислялось как среднее арифметическое величин  $r_{iT}$  и  $r_{jT}$ . Однако в действительности, предложенное определение времени перехода по сети являлось эквивалентным введённым ранее понятиям. К данной модели обращались при построении маршрутов банковских курьеров в ряде города США (Нью-Йорк, Балтимор, Детройт, Сиэтл), однако деталей и результатов интеграции модели в статье указано не было.

Основная неточность ранних моделей TD-VRP заключалась в том, что модели не удовлетворяли свойству FIFO (First In, First Output). Это интуитивно понятное правило, которое необходимо учитывать при моделировании движения транспорта. FIFO-свойство гарантирует, если машина в момент времени  $t$  выезжает из узла сети  $i$  и следует в узел  $j$ , то идентичная машина, отправляющаяся по тому же маршруту в момент времени  $t + e$ , где  $e > 0$ , всегда приедет позже первой. Для выполнения данного свойства в работе [14] было предложено делать перерывы в движении машин для сглаживания функции

времени перехода, заставляя ожидать транспортные средства в узлах сети. При этом очевидно, что описанное решение проблемы вряд ли могло быть осуществлено на практике. Удовлетворяет свойству FIFO модель TD-VRP, рассмотренная в статье [16]. Горизонт планирования в данной модели разделён на периоды, и время перехода, определяемое непрерывной линейной функцией, изменяется от периода к периоду. Время перехода от узла  $i$  сети до узла  $j$  вычисляется с помощью заданной для каждого периода скоростью движения и расстоянием  $r_{ij}$  между узлами. Расстояние  $r_{ij}$  является постоянным для всех периодов, что является недостатком данного метода, так как не учитываются возможное изменение пути проезда от узла  $i$  до  $j$  в случае возникновения заторов на маршруте. Также, в данной работе функция времени перехода является линейной, и она не может корректно отразить реальную ситуацию на улично-дорожной сети. Для решения этой проблемы в статье [17] была представлена полиномиальная функция времени перехода. Однако предложенная постановка не гарантирует выполнения FIFO-свойства и требует большой вычислительной мощности даже для задач малой размерности.

Метаэвристическая процедура GRASP, описанная в работе [17], была позже применена для задач на УДС города Турино, Италия. Данные о сети предоставила организация «Intelligent Transport Systems and Infomobility society». Скорости движения по главным улицам города были вычислены как среднее арифметическое данных, полученных за 2 недели ежедневного мониторинга электронной системой с 7:20 до 20:00. Интервал между 2 последовательными измерениями равнялся 5 минутам. Данные по маленьким улицам, по которым не удалось собрать информацию о трафике, полагались постоянными в течение всего периода маршрутизации. Рассмотренный в статье при-

мер содержит только 10 точек на сети города, однако отмечается существенное преимущество TD-VRP модели в генерации маршрутных планов меньшей продолжительности перед классической постановкой задачи VRP. Работа [18] описывает эксперименты, проведённые в городе Берлин, Германия. Информация для задачи маршрутизации была получена из информационно-транспортной службы «LISB». Время перехода было доступно для каждого ребра улично-дорожной сети для 214 пятиминутных интервалов. Средняя скорость движения по городу Берлин составила 32.9 км/ч со стандартным отклонением 11 км/ч. Размер тестовых задач маршрутизации варьировался от 58 клиентов и 8 машин до 786 клиентов и 84 машин. Задачи были решены эвристиками последовательных вставок, локального поиска и алгоритмом Кларка-Райта.

Алгоритм муравьиный колонии решения задачи TDVRPTW (Time Dependent Vehicle Routing Problem with Time Windows) был рассмотрен в работе [19] в применении к УДС города Падуя, Италия. Информация о времени перехода по сети города из 1522 узлов и 2579 рёбер была собрана автоматической системой контроля за трафиком «Cartesio». Измерения проводилась для всех рёбер только в часы пик, а для 50 из них — каждый час. В тестовых задачах необходимо было посетить 30 клиентов, используя 10 доступных в депо машин. TD-решения уменьшили среднюю длительность маршрутов по сравнению с моделью CS (Constant Speed) на 7,58%.

Возможность интегрирования TD-моделей в структуру компаний была также рассмотрена в ряде работ. Например, благодаря использованию модели TDVRP крупнейшая сеть электронных товаров Тайваня 3С сократила время выполнения работ по транспортировке товаров на 22.26% по сравнению с те-

кущей стратегией компании [20]. Задача включала в себя 25 розничных магазинов и 5 периодов дня с 8:00 утра до 19:00 вечера. Авторы работы отмечали, что сокращение времени маршрутов влечёт за собой увеличение пройденного расстояния машинами компании, и соответственно возникают дополнительные расходы на топливо. Как итог, для дальнейших исследований авторы предложили рассматривать эти два фактора в совокупности.

Задача TD-маршрутизации была так же решена на юго-западе Великобритании при построении маршрутных планов для компании оптовой торговли электроникой [21, 22]. Количество клиентов в день варьировалось от 40 до 64 при 7 доступных машинах.

В последние годы актуальным становится класс задач, который рассматривает TD-маршрутизацию с целевой функцией, зависящей одновременно от времени в пути и от объёма выхлопов  $CO_2$  в окружающую среду [22, 23]. Также TD-модели были реализованы для решения модификаций классической постановки задачи VRP: с вывозом и доставкой товара [24] и с учётом хранения ресурсов [25]. Подробный обзор математических моделей задач маршрутизации с переменными значениями времени перехода между узлами сети представлен в [26].

Только в некоторых работах исследователи для валидации полученной модели используют тестовые задачи из стандартных библиотек, при этом часто изменяя тестовые задачи. Например, в статье [16] рассмотрена библиотека Соломона [27], но авторы не учитывают заданные ограничения на грузоподъёмность машин и общий объём парка в депо. Эксперименты, описанные в [19], также проводились на тестовых задачах данной библиотеки, однако результаты невозможно сравнить, так как была использована отличная от преды-

дущей работы функция времени перехода по рёбрам сети. Новые тестовые задачи представлены в [28]. Они более точно отражают особенности улично-дорожной сети в условиях пробок и загруженности. Сравнение с предыдущими работами на 56 тестовых задачах библиотеки Соломона было проведено авторами в статье [29], где использовался генетический алгоритм решения задачи TDVRPTW.

## 1.2 Постановка задачи TD-MDVRP

Рассмотрим граф  $G = (V, E)$ , где  $V = \{1, \dots, N + M\}$  — множество вершин графа  $G$ , а  $E = \{(i, j) \mid i, j \in V\}$  — множество ребер данного графа. Множество вершин состоит из множества клиентов  $V_{cust}$  и множества депо  $V_{depot}$ , при этом  $V = V_{cust} \cup V_{depot}$ , где  $V_{cust} = \{1, \dots, N\}$  и  $V_{depot} = \{1, \dots, M\}$ .

Имеется множество  $K = \{1, \dots, L\}$  машин. Грузоподъемность машины  $k$  обозначим через  $Q_k$ . Положим, что одна машина может совершить только одну поездку (с возвращением в депо), при этом суммарный объём перевозимый машиной  $k$  груза, не должен превышать значение  $Q_k$ .

Каждый клиент  $j \in V_{cust}$  имеет спрос  $d_j$  в товаре, который должен быть удовлетворён. Будем считать, каждый клиент может быть посещён только одной машиной. Количество товара в депо (на складе) достаточно для удовлетворения спроса всех клиентов тестовой задачи.

Будем рассматривать задачу маршрутизации с матрицей переменных значений времени перехода между клиентами. Это означает, что время в пути от вершины  $i$  до  $j$ , где  $(i, j) \in E$ , будет меняться в течение периода маршрутизации. Каждому ребру  $(i, j) \in E$  сопоставим функцию  $t_{ij}(b_i)$  времени перехода по нему. Значение  $b_i$  соответствует времени отправления от клиента  $i$ . Через  $b_0^v$  обозначим время выезда из депо для

машины  $v$ , где  $v \in K$ . Общий вид функции времени перехода  $t_{ij}(b_i)$  от клиента  $i$  до  $j$  будет рассмотрен далее.

Целью оптимизации будет являться нахождение маршрутного плана для машин (т.е. множество маршрутов для каждой из машин в депо), который удовлетворяет спросу клиентов и не нарушает наложенных на переменные ограничения. При этом время суммарное время всех машин в пути должно быть минимально.

Описанная выше задача TD-MDVRP может быть сформулирована в рамках целочисленного программирования. Пусть  $\{x_{ij}\}_{i,j \in V}$  — булевы переменные, которые равны единице, если в маршрутном плане клиент  $i$  посещается сразу за клиентом  $j$ , и ноль — иначе. Введём бинарные переменные  $\{y_{ik}\}_{i \in V, k \in K}$ , которые показывают принадлежность клиента  $i$  маршрутному плану машины  $k$ . Тогда задачу можно представить в следующем виде:

$$\sum_{i \in V} \sum_{j \in V} t_{ij}(b_i) x_{ij} \rightarrow \min, \quad (1)$$

при ограничениях:

$$\sum_{i \in V} x_{ij} = \sum_{i \in V} x_{ji} \quad \forall j \in V; \quad (2)$$

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V_{cust}; \quad (3)$$

$$\sum_{i \in V_{cust}} d_i y_{ik} \leq Q_k \quad \forall k \in K; \quad (4)$$

$$b_j \geq \sum_{i \in V_{cust}} x_{ij} (b_i + t_{ij}(b_i)) + x_{0j} \sum_{v \in K} y_{jv} (b_0^v + t_{ij}(b_0^v)), \quad \forall j \in V_{cust} \quad (5)$$

Целевая функция (1) представляет собой суммарное время движения всех машин согласно построенному маршрутному плану.

Ограничения (2) гарантируют непрерывность маршрутов. Ограничения (3) показывают, что каждый клиент должен быть посещён только один раз.

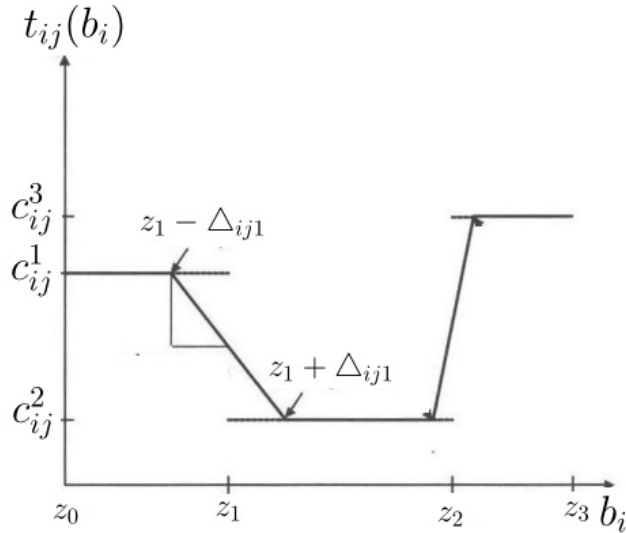
Так как данная группа ограничений учитывает, что из одного депо могут выезжать сразу несколько машин, равенства (3) верны только для значений  $j \in V_{cust}$ . Неравенства (4) накладывают ограничения на грузоподъёмность машин. Последняя группа ограничений (5) представляет нижние границы времени отправления от каждого клиента.

Как было определено выше, момент времени  $b_i, i \in N$  соответствует времени отправления от клиента  $i$ . Будем полагать, что непрерывная кусочно-линейная функция  $t_{ij}(b_i)$  имеет вид:

$$t_{ij}(b_i) = \begin{cases} c_{ij}^k, & z_{k-1} + \Delta_{ij,k-1} \leq b_i \leq z_k - \Delta_{ijk} \\ c_{ij}^k + \frac{(c_{ij}^{k+1} - c_{ij}^k)(b_i - z_k + \Delta_{ijk})}{2\Delta_{ijk}}, & z_k - \Delta_{ijk} < b_i < z_k + \Delta_{ijk}, \end{cases} \quad (6)$$

при  $k = 1, \dots, K; \Delta_{ij0} = \Delta_{ijK} = 0$ .

Впервые данная модель была представлена в работе [18]. Общий вид функции  $t_{ij}(b_i)$  представлен на рисунке 1.



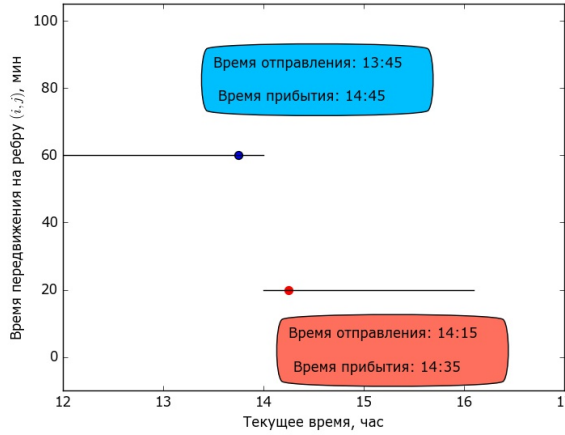
**Рис. 1.** Функция времени перехода по ребру  $(i, j)$

Также в статье [18] была доказана лемма, что для справедливости FIFO-

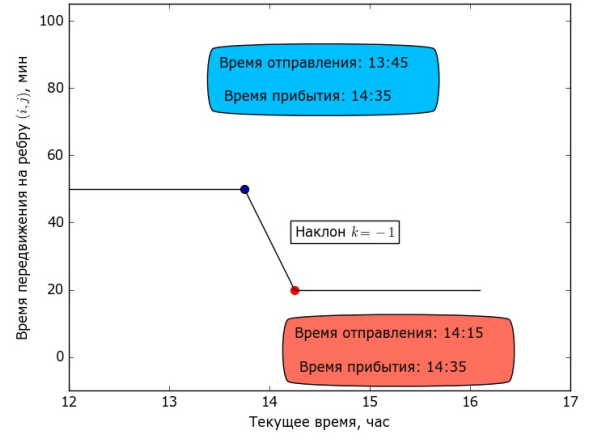
свойства и непрерывности матрицы  $T(b) = b + t_{ij}(b)$  достаточно выполнения следующих условий:

1.  $\Delta_{ijk} > 0, \quad k = 1, \dots, K - 1.$
2.  $z_{k-1} + \Delta_{ij,k-1} \leq z_k - \Delta_{ijk}, \quad k = 1, \dots, K.$
3.  $(c_{ij}^{k+1} - c_{ij}^k) / 2\Delta_{ijk} > -1, \quad k = 1, \dots, K - 1.$

Первые два условия в лемме очевидны; третье означает, что угол наклона функции времени перехода  $t_{ij}(b_i)$  по ребру  $(i, j)$  на интервале  $(z_k - \Delta_{ijk}, z_k + \Delta_{ijk})$  должен быть больше 45 градусов [20]. Для наглядности рассмотрим следующий пример. На рисунке 2 представлен график функции времени перехода по ребру  $(i, j)$ .



**Рис. 2.** Функция  $t_{i,j}(b_i)$ ,  
неудовлетворяющая свойству FIFO



**Рис. 3.** Функция  $t_{i,j}(b_i)$ ,  
удовлетворяющая свойству FIFO

По оси X отмечено время начала движения по рассматриваемому ребру, по оси Y — время в минутах для проезда по нему. Если машина отправляется из вершины  $i$  в 13:45, то она приедет в вершину  $j$  в 14:45. Однако, если машина отправится из вершины  $i$  позже, а именно в 14:15, согласно функции времени перехода она окажется в вершине  $j$  в 14:35. Очевидно, что наруша-



ется свойство FIFO, рассмотренное выше.

Для выполнения FIFO-свойства необходимо, чтобы угол наклона линейной функции перехода был больше  $-1$ . На рисунке 3 последнее условие леммы выполнено. Каждая машина, отправляющая из вершины  $i$  до вершины  $j$  в период с 13:45 до 14:15, закончит движение по данному ребру в 14:35.

### 1.3 Эвристические алгоритмы решения задачи MDVRP

Точные методы редко используют для решения задач большой размерности на практике ввиду их NP-полноты. По этой причине обычно применяют эвристические алгоритмы, которые не гарантируют оптимальность решения, но дают в целом приемлемые результаты за время, существенно меньшее по сравнению с точными методами.

Среди эвристических алгоритмов решения задач транспортной маршрутизации можно выделить два больших класса: конструктивные (англ.: *tour construction heuristics*) и итерационные (англ.: *tour improvement heuristics*).

Конструктивные эвристические алгоритмы вычисляют единственный маршрут для рассматриваемой задачи и не улучшают его в дальнейшем. Наиболее известными эвристиками среди них являются семейство жадных алгоритмов и алгоритмов ближайшего соседа, а также алгоритм Кристофидеса и различные модификации методов со вставками. Указанные методы генерируют допустимое решение, последовательно вставляя вершины в текущий маршрут, пока итоговый маршрутный план не будет составлен. Таким образом, часть маршрутного плана, сгенерированная к настоящему времени, не изменится и в итоговом решении.

Второй класс эвристических подходов, а именно итерационные эвристики, использует уже готовые маршруты, полученные одной из конструктивных

эвристик, последовательно улучшая их на каждой итерации. Среди итерационных алгоритмов выделим отдельный класс методов, называемых метаэвристиками. Это достаточно общие итерационные процедуры, включающие в себя рандомизацию, элементы самообучения, интенсификацию и диверсификацию поиска, адаптивные механизмы управления, а так же в некоторых случаях — конструктивные эвристики и методы локального поиска. Метаэвристики принято разделять на траекторные методы, в них на каждой итерации имеется одно допустимое решение и осуществляется переход к следующему, и на алгоритмы, работающие сразу с популяцией решений. Среди алгоритмов первой группы выделим методы имитации отжига, поиск с запретами, поиск с чередующимися окрестностями. Ко второй группе относят — генетические эвристики; алгоритмы, основанные на моделировании поведения муравьиной колонии, а также вероятностные жадные алгоритмы и некоторые другие [30].

По сравнению с классической постановкой задачи маршрутизации VRP, малое число работ исследователей посвящено её модификации — задаче MDVRP и эвристическим алгоритмам решения. В работе [31] авторы предложили подход, который в дальнейшем стал широко использоваться для решения практических задач MDVRP. Была рассмотрена стратегия «кластеризация и маршрутизация», описаны 6 методов разбиения множества клиентов для их назначения на обслуживание одному из депо, проведены вычислительные эксперименты. При постановке задачи MDVRP многие исследователи делали ряд допущений, например: одинаковая грузоподъемность транспортных средств, неограниченное число транспортных средств в депо и другие. Математическая формулировка задачи MDVRP, представленная в [32]

учла данные допущения. Обе части задачи (кластеризация и маршрутизация) были решены алгоритмом имитации отжига. Метод имитации отжига является рандомизированным алгоритмом локального поиска. По аналогии с физическим процессом имитации отжига, в котором после отжига требуется снижать температуру для восстановления твёрдого состояния с низкой энергией, в данном алгоритме применяют управляющий параметр, называемый температурой, для вычисления вероятности принятия решений, не улучшающих значение целевой функции.

Другой эвристический алгоритм, поиск с запретами, был применён в работе [33], при решении модификации MDVRP. Рассматриваемая модель MDVRP позволяет транспортным средствам загрузить дополнительный товар на промежуточных складах вдоль построенного маршрута. Особенности алгоритма поиска с запретами являются возможность перехода к другому решению даже в случае, если это приведёт к увеличению целевой функции и построение нового решения с учётом запрета на включение части некоторой окрестности текущего решения [34]. Эксперименты в данной работе проводились на случайно сгенерированных тестовых задачах.

В более поздних работах вместо двух этапов решения задачи MDVRP выделяют три: кластеризация, маршрутизация и планирование. Под планированием понимают построение маршрутов для движения, а под маршрутизацией — распределение клиентов между транспортными средствами. Для решения последней задачи используют, например, алгоритм Кларка-Райта [35]. Итоговый маршрутный план в данной статье был получен генетической эвристикой. Тестовые задачи рассматривались из стандартной библиотеки Cordeau's Library [36]. Было произведено сравнение с текущими работами и

показано, что генетический алгоритм позволяет эффективно решать задачу MDVRP.

#### 1.4 Генетический алгоритм решения задачи MDVRP

Для решения задачи TD-MDVRP в данной работе был выбран генетический алгоритм. Генетический алгоритм — одна из самых популярных метаэвристик, моделирующая в своей работе операции, присущие процессам, которые протекают в природе в рамках естественного отбора. Первым, кто исследовал генетический алгоритм, был американский учёный Джон Холланд, в 1975 году [37].

Для генерации маршрутного плана тестовой задачи MDVRP необходимо правильно задать условия для каждого из этапов эволюционного алгоритма.

Классический генетический алгоритм состоит из следующего набора операций:

- 1) Инициализация популяции
- 2) Скрещивание
- 3) Мутация
- 4) Оценка приспособленности и селекция

Введём следующие обозначения:  $P(i)$  — популяция на  $i$ -ом поколении,  $Q(i)$  — потомство на  $i$ -ом поколении,  $k_{max}$  — количество поколений, генерируемых для получения одного решения,  $\lambda$  — размер популяции,  $\mu$  — размер популяции потомства,  $\beta(d)$  — «полный маршрут» клиентов для депо  $d$ .

В рамках работы была реализована генетическая эвристика для генерации маршрутных планов решения задачи TD-MDVRP. Для реализации программы использовался язык программирования Java. Краткое описание ра-

боты выбранной метаэвристики представлено в алгоритме 1. Далее будут детально рассмотрены каждый из этапов генетической эвристики.

---

**Алгоритм 1:** Генетический алгоритм решения задачи TD-MDVRP

---

- 1: Инициализация начальной популяции  $P(0)$ ,  $|P(0)| = \lambda$ ,  $k = 0$ .
  - 2: **Пока**  $k < k_{max}$ , **выполнять:**
  - 3:      $Q(k) = \emptyset$ .
  - 4:     **Пока**  $|Q(k)| < \mu$ , **выполнять:**
  - 5:          $b = \text{Кроссовер}(\text{Селекция}(P(k)), \text{Селекция}(P(k)))$
  - 6:          $\text{Мутация}(b)$
  - 7:          $Q(k) = Q(k) \cup b$
  - 8:     **конец цикла**
  - 9:     **Локальное улучшение**  $(P(k) \cup Q(k))$
  - 10: **конец цикла**
  - 11: **Формирование** нового поколения  $P(k + 1) = \text{Селекция}(P(k) \cup Q(k))$ .
  - 12:  $k = k + 1$
  - 13: **конец**
- 

На начальном этапе эвристика вычисляет исходную популяцию особей, каждая из которых представляет собой допустимое решение задачи. Особь в популяции задаётся уникальным набором генов. В хромосоме особи записан порядок обхода городов, соответствующий одному допустимому решению. Таким образом, хромосома  $x$  для тестовой задачи с  $n$  вершинами и 1 транспортным средством принимает вид  $x^{(n)} = v_1 v_2 v_3 \dots v_{n-1} v_n v_1$ , где  $v_i$  – вершины графа задачи. В ходе каждой итерации хромосома оценивается с использованием функции соответствия (англ.: *fitness function*). В случае задачи TD-MDVRP

функция соответствия принимает значение времени объезда всех клиентов согласно маршрутному плану, закодированному в хромосоме особи.

*Пример.* Маршруты движения 1—5—3—6—1 и 1—4—2—1 для тестовой задачи с 5 клиентами и 2 транспортными средствами соответствует хромосоме  $x^{(6)} = \langle v_1 v_5 v_3 v_6 v_1 v_4 v_2 v_1 \rangle$  и матрице переходов  $X$ .

$$X = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Элемент матрицы  $x_{ij}$  равен 1, если хромосома  $x^{(6)}$  особи содержит ребро  $(v_i, v_j)$ , а иначе — нулю.

Исходная популяция генетического алгоритма может быть сформирована как конструктивной эвристикой, так и случайным образом. Хороший выбор начальной популяции сокращает время достижения глобального оптимума. С одной стороны, в начальной популяции должно наблюдаться разнообразие особей, так как от этого зависит, насколько полноценно мы исследуем пространство допустимых решений задачи. С другой — требуется наличие достаточного количества экономичных маршрутов, чтобы процесс схождения к глобальному оптимуму алгоритм происходил быстрее. В программной реализации треть исходной популяции формировалась случайным образом для того, чтобы поддерживать генетическое разнообразие в популяции, а остальные особи генерировались по эвристике «полного тура». Псевдокод алгоритма 2 описывает общую схему данной эвристики.

---

**Алгоритм 2:** Инициализация начальной популяции ГА

---

- 1: Для всех  $d \in V_{dep}$ ,  $\beta(d) = 0$
  - 2: Для всех  $c \in V_{cust}$ , **выполнять:**
  - 3:      $d =$  Ближайшее Депо (Случайный выбор ( $c$ ))
  - 4:     **Вставить** клиента  $c$  в большой маршрут  $\beta(d)$
  - 5: **конец цикла**
  - 6: Для всех  $d \in V_{dep}$  :
  - 7:     **Разделить**  $\beta(d)$  на маршруты для каждой машины.
  - 8: **конец цикла**
- 

В данной эвристике множество всех клиентов разбивается на кластеры в зависимости от близости к одному из депо. Далее для каждого из депо строится «полный маршрут» из клиентов, отнесённых к данному депо на предыдущем шаге алгоритма. Новый клиент вставляется в «полный маршрут» с помощью эвристики вставок: для клиента эвристика выбирает такое место в текущем туре, чтобы затраты на вставку были минимальны. На рисунке 4а красным цветом обозначены клиенты, одного из которых необходимо поместить в «полный тур» заданного депо. Минимальная вставка представлена на рисунке 4б, она отмечена пунктирной линией.

После такого как будут сформированы «полный туры» для каждого депо, их следует разделить на несколько частей в соответствии с ограничениями задачи и каждый под-маршрут назначить на обслуживание одному транспортному средству. Клиенты добавляются в маршрутные планы по очереди, как они следовали в «полном туре» для депо, пока удовлетворяются все ограничения. Если ограничения нарушаются, то данный клиент назнача-



**Рис. 4.** Эвристика вставок при инициализации начальной популяции

ется следующему транспортному средству в депо. Также допустимо разбиение множества клиентов из «полного тура» на равные части между транспортными средствами в депо.

На следующем этапе генетического алгоритма между особями в популяции происходит скрещивание. В скрещивании принимают участие две особи, и они генерируют потомков с помощью методов комбинирования и наследования особенностей обоих родителей. Главная цель операции скрещивания — улучшить качественный признак потомственной особи и всей популяции в целом. Приведём некоторые варианты скрещивания, которые могут быть применены для решения задач транспортной маршрутизации [38, 39].

Самым простым видом скрещивания является частичное скрещивание [40]. В двух родительских особях случайным образом выбираются точки для скрещивания, делящие хромосомы на некоторое количество частей. Потомственные особи наследуют гены родителей, в которых изменён порядок следования частей. Другой метод проведения операции скрещивания называется циклическое скрещивание (англ.: *cycle crossover*) [41]. Суть этого метода заключается в том, что в каждом гене потомственной хромосомы может рас-



полагаться вершина только из того же гена одной из родительских особей.

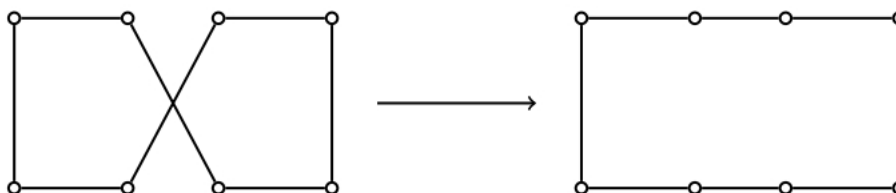
Для использования в программной реализации решения задачи TD-MDVRP было выбран кроссовер перестановки рёбер (англ.: *edge recombination crossover*) [42]. Данный оператор интерпретирует хромосому как неориентированный цикл. Основная идея кроссовера перестановки рёбер заключается в том, что вероятность наследования ребра потомственной особью зависит от вхождения данного ребра в хромосомы родителей. Вероятность вхождения будет наибольшая, если ребро присутствует у обоих родителей и наоборот. Процедура формирования гена потомка связана с понятием «соседства». Соседом текущей вершины будем называть вершину, смежную данной в первой и второй хромосомах родителей. На каждом шаге эвристики следующая вершина выбирается из множества соседей предыдущей. Если существует несколько допустимых вершин для вставки в хромосому потомка, алгоритм выберет ту, мощность множества соседей которой наименьшая. Если множество допустимых соседей пусто, необходимо выбрать случайную вершину из множества ещё не использованных. Рассмотрим две родительские особи:  $parent_1 = 123456789$  и  $parent_2 = 412876935$ . Начнём формирование маршрута с вершины 1. Следующую за ней вершину следует выбирать из множества  $N_1 = \{2, 4, 9\}$ . Рассмотрим мощности соседей множества  $N_1$ , получим что вершина 2 имеет 3 соседа, вершина 3 имеет так же 3 соседа, а вершина 9 – 4. Наименьшая мощность будет у вершин 2 и 4, случайным образом выберем любую из них и поместим в ген потомка. Имеем после первого шага алгоритма  $child = 14*****$ . Продолжая формировать потомственную особь по тому же правилу, получим следующий результат:  $child = 145678239$ . В применении кроссовера перестановки рёбер к TD-MDVRP, вершины тестовой задачи,

обозначающие депо, отмечаются отдельно при формировании множества соседей каждой вершины. Это необходимо, чтобы в итогом цикле отсутствовали рёбра, соединяющие два депо.

Третьим этапом генетической эвристики является мутация. По сравнению с операцией скрещивания, в которой принимают участие сразу две особи популяции, для мутации необходима только одна. Данная операция требуется для восстановления особей, которые выпали из текущей популяции, а также во время неё формируются гены, ранее не представленные в популяции. Оператор мутации позволяет переназначать обслуживание клиента другим транспортным средствам. Случайно выбранный клиент удаляется из маршрутного плана одного транспортного средства и с помощью эвристики минимальной вставки алгоритм находит новое место для данного клиента среди всех текущих маршрутов. Далее алгоритм производит оценку функции приспособленности всех особей текущей популяции. Если после мутации значение функции приспособленности рассматриваемой хромосомы стало меньше, чем до операции, новая особь вытесняет из популяции исходную. Рассмотрим понятие уровня мутации (англ.: *mutation rate*). Под уровнем мутации будем понимать долю всей особей от общего числа, которые подвергаются мутации. В программной реализации значение уровня мутации задавалось равным 33%. Малое значение рассматриваемого параметра приводит к ситуации, когда большинство генов приспособленных особей не будут рассмотрены. Однако большой уровень мутации ведёт к частым случайным возмущениям.

На этапе локального улучшения будем использовать эвристику 2-окружения. Алгоритм 2-окружения удаляет два случайных несмежных ребра из маршрута обхода клиентов одного транспортного средства и соединяет

две части маршрута другим способом. Очевидно, что для вставки 2 новых рёбер возможен только один вариант, чтобы полученное после данной операции решение осталось допустимым, т.е. являлось бы гамильтоновым циклом. Пример операции 2-окружение продемонстрирован на рисунке 5.



**Рис. 5.** Пример операции 2-окружение

После операций скрещивания, мутации и локального улучшения происходит оценка функции приспособленности особей из популяции. Неприспособленные особи, т.е. особи с наибольшим значением целевой функции (длины или времени) удаляются из популяции и не участвуют в эволюционном процессе далее. В программной реализации процент особей, которые переходят от предыдущего поколения к следующему, был задан равным 75%.

Процесс эволюции генетической эвристики, как правило, останавливают согласно следующим критериям: заданному числу поколений; времени, затрачиваемому на эволюцию; схождению популяции. При схождении популяции наблюдается остановка в эволюционном развитии особей, т.е. имеет место процесс стагнации. Основная причина стагнации — попадание в локальные оптимумы, так как при этом происходит потеря разнообразия генов эволюционного процесса.

### 1.5 Динамическая адаптация генетического алгоритма

Как известно из работы [43] одним из новых критериев оценки эффективности эвристических алгоритмов является временная состоятельность генерируемых этим алгоритмом решений. Оценка уровня временной состоятельности эвристик зависит от временной состоятельности (или динамической устойчивости) генерируемых решений и рассчитывается в ходе численных экспериментов [44, 45]. Свойство временной состоятельности (динамической устойчивости) решений динамических систем исследовалось еще Ричардом Беллманом [46] и состоит в том, что решения, оптимальные на всем промежутке исследования динамической системы, сохраняют свою оптимальность и в текущих задачах, рассматриваемых на любых более коротких промежутках.

Понятие временной состоятельности решения в рамках задач транспортной маршрутизации означает, что рассматриваемый маршрутный план удовлетворяет свойству оптимальности Беллмана в каждой из подзадач в ходе реализации исходно сгенерированного маршрута. Очевидно, что данное условие будет выполнено только в случае оптимальности данного решения, т.е. оно должно быть вычислено точными алгоритмами. При этом эвристические подходы будут формировать, как правило, несостоятельные во времени решения.

Каждое решение  $s(p)$  задачи транспортной маршрутизации представляет собой порядок обхода вершин в течение заданного количества периодов, равного  $T$ . Через  $s^+(t, p)$  в формуле (7) обозначим часть итогового маршрута  $s(p)$ , которая включает в себя вершины в построенном маршрутном плане, посещаемые после периода  $t$ , а через  $s^-(t, p)$  — вершины, посещённые за периоды  $\tau = 0, 1, \dots, t$ .

Будем рассматривать текущую задачу  $p(s^-(t, p))$ , отличную от начальной задачи  $p$  тем, что из множества вершин удалены вершины, входящие в часть маршрута  $s^-(t, p)$ . Тогда  $s(p(s^-(t, p)))$  будет означать решение текущей задачи, полученной заданной изначально эвристикой.

**Определение.** Маршрут  $s(p)$  *состоятелен во времени*, если  $\forall t = 1, \dots, T - 1$  справедливо неравенство:

$$f(s^+(t, p)) \leq f(s(p(s^-(t, p)))), \quad (7)$$

где  $f$  — целевая функция для рассматриваемой задачи маршрутизации транспорта.

Пусть задано множество тестовых задач маршрутизации транспорта  $P$  и  $S(p)$  — конечное множество решений, сгенерированных эвристикой  $H$  для тестовой задачи  $p \in P$ . Введём величину  $b(s(p), t)$ , соответствующую количеству запусков, при которых исходное решение  $s(p) \in N$  перестало быть состоятельным во времени после периода  $t$ .

**Определение.** *Экспериментальный уровень временной состоятельности эвристики  $H$*  является величиной, определяемой по следующей формуле:

$$conH = 1 - \frac{1}{M|P|} \sum_{p \in P} \frac{1}{|S(p)|} \sum_{s(p) \in N} \sum_t^{T-1} b(s(p), t), \quad (8)$$

Стоит отметить, что  $0 \leq conH \leq 1$ . Таким образом, чем ближе значение  $conH$  к 0, тем больше вероятность, что решение, сгенерированного данной эвристикой, потеряет оптимальность в ходе реализации маршрутного плана. А для маршрутов, полученных точными алгоритмами, значение данной оценки равно 1.

В работе [47] была проведена оценка экспериментального уровня временной состоятельности генетического алгоритма на тестовых задачах от-

крытой библиотеки TSPLib[48]. Среднее значение экспериментального уровня временной состоятельности генетической эвристики в применении к решению TSP составило 0,276.

$$conGA = 0,276$$

Рассчитанное значение оценки получилось достаточно низким: только около четверти исходно полученных решений сохраняют свойство оптимальности в процессе своей реализации. Таким образом, на каждом из периодов существуют маршруты, с меньшими значениями целевой функции по сравнению с исходно сгенерированным маршрутным планом.

---

**Алгоритм 3:** Динамическая адаптация эвристич. алгоритма

---

- 0: Генерация  $N$  решений эвристикой  $H$  для  $p \in P$ .
  - 1: Вычислим  $s_1(p) = \arg \min_{s(p) \in N} f(s(p))$ .
  - 2: **От**  $t = 1$  **до**  $T - 1$  **выполняем**
  - 3:     Формирование текущей задачи  $p(s_t^-(t, p))$ , имеем  $N$  решений.
  - 4:     Вычислим  $s_{t+1}^*(p) = \arg \min_{s(p(s_t^-(t, p))) \in N} f(s(p(s_t^-(t, p))))$ .
  - 5:     Проверка свойства временной состоятельности для  $s_t(p)$ .  
       **Если**  $f(s_t^+(t, p)) > f(s_{t+1}^*(p))$  **тогда**
  - 6:         Изменим маршрут обхода на  $s_{t+1}(p) = s_t^-(t, p) \cup s_{t+1}^*(p)$ .
  - 7:     **конец**
  - 8: **конец**
- 

В работе [49] авторы отмечают, что процесс поиска оптимального решения в генетических алгоритмах направляется исключительно полученными значениями целевой функции в предыдущих точках пространства решений. При этом никак не используются предположения о таких свойствах целевой

функции и ограничений как выпуклость, дифференцируемость, выполнение соотношения оптимальности Беллмана. Алгоритм динамического улучшения эвристических методов решения задач маршрутизации транспорта был исследован в работе [50]. Понятие временной состоятельности решения, составляющее основу процедуры динамического улучшения эвристических алгоритмов, было подробно изучено за последние годы в области теории игр [51, 52, 53].

Общая схема динамической адаптации эвристического алгоритма  $H$  для решения задачи TSP представлена в алгоритме 3 [47].

Рассмотренную процедуру динамической адаптации будем применять в экспериментах в главе 3 при генерации маршрутных планов задачи TD-MDVRP. Также будет представлено сравнение классического генетического алгоритма и его динамической адаптации.

## 2 Задача распределения транспортных потоков

### 2.1 Постановка задачи распределения транспортных потоков

Из работы [54] известно, что любая транспортная система по истечении некоторого времени приходит в равновесное состояние. Первый принцип Вардропы, сформулированный в 1952 г. гласит что, «время передвижения по всем используемым маршрутам одинаково для всех участников движения и меньше времени, которое потратит любой участник движения, изменив свой маршрут». Каждый участник независимо от остальных выбирает маршрут для передвижения, с целью снизить свои затраты. Распределение потока по сети, соответствующее рассмотренному принципу, называется конкурентным равновесием. Оно достигается, когда никто из участников движения не может снизить свои расходы на перемещение.

Будем рассматривать транспортную сеть в виде ориентированного графа  $G = (N, A)$ , где через  $N$  обозначим множество последовательно пронумерованных узлов, а через  $A$  — множество последовательно пронумерованных дуг данного графа  $G$ . Пусть  $R$  множество узлов, которые соответствуют районам отправления,  $S$  — районам прибытия. Введём величину  $K_{rs}$ , обозначающую множество маршрутов между районом отправления  $r \in R$  и районом прибытия  $s \in S$ ;  $x_a$  — транспортный поток по дуге  $a \in A$ ;  $d_a$  — время передвижения по дуге  $a \in A$ . Транспортный поток по отдельному маршруту  $k \in K_{rs}$  обозначим через  $f_k^{rs}$ , а совокупный транспортный спрос между  $r \in R$  и  $s \in S$  —  $F^{rs}$ . Введем булеву функцию, которая показывает принадлежность



ребра  $a$  маршруту  $k \in K_{rs}$ :

$$\delta_{ak}^{rs} = \begin{cases} 1, & \text{если дуга } a \text{ включена в маршрут } k \in K_{rs}, \\ 0, & \text{иначе.} \end{cases}$$

Математическая формализация первого принципа Вардропа возможна в виде задачи минимизации с ограничениями:

$$\sum_{k \in K_{rs}} f_k^{rs} = F^{rs}, \quad \forall r \in R, s \in S, \quad (9)$$

$$f_k^{rs} \geq 0, \quad \forall k \in K_{rs}, r \in R, s \in S, \quad (10)$$

$$x_a = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs}, \quad \forall a \in A. \quad (11)$$

Целевая функция, соответствующую конкурентному равновесию на сети, имеет следующий вид:

$$\min_x \sum_{a \in A} \int_0^{x_a} d_a(u) du,$$

где  $d_a(f)$  — функция задержки потока объёма  $f$  на ребре  $a \in A$ . В качестве функции задержки выберем BPR-функцию:

$$d_i(f_i) = t_i^0 \left( 1 + \alpha_i \left( \frac{f_i}{p_i} \right)^{\beta_i} \right),$$

где  $t_i^0$  — время свободного движения по ребру  $i$ ,  $p_i$  — пропускная способность ребра  $i$ . Задача минимизации при ограничениях (1)–(3) примет вид

$$T(f) = \min_f \sum_{i=1}^{|A|} \int_0^{f_i} t_i^0 \left( 1 + \frac{u}{p_i} \right) du.$$

Решение задачи распределения потоков (ЗРП) будем искать с помощью алгоритма Франка – Вульфа, широко применяемого в задачах нелинейной оптимизации [55].

1. Обозначим допустимое решение этой задачи как вектор  $f^0$ . Решаем задачу линейного программирования  $\min T(f)$ ,

$$T(f) = T(f^k) + \nabla T(f^k)(f - f^k),$$

при ограничениях (1)–(3). Стоит заметить,  $\frac{\partial Z(f^k)}{\partial f_a} = d_a(f_a^k)$ .

2. Пусть  $z^k$  — решение задачи ЛП из первого шага, тогда  $p^k = z^k - f^k$ . Находим длину шага  $l_k$ :  $\min T(f^k + l_k p^k)$ , где  $0 \leq l_k \leq 1$ . Вычислим  $f^{k+1} = f^k + l_k p^k$ .
3. Проверяем условие останова алгоритма:  $|T(f^{k+1}) - T(f^k)| < \varepsilon$ .

После решения задачи ЗРП получим вектор  $f$  распределения транспортного потока по рёбрам ориентированного графа  $G$ . Далее вычислим время передвижения по любому ребру  $a \in A$ , используя заданные функции задержки. Для этого применим алгоритм Дейкстры, находящий кратчайшие пути на графе от одной вершины до любой другой [56]. Схема работы алгоритма для вершины  $a$  приведена ниже.

1. Каждой вершине соответствует метка, означающая наименьшее известное расстояние от данной вершины до  $a$ . Изначально примем метку самой вершины  $a$  равной 0, метки других вершин – бесконечности.
2. Из множества еще не посещенных вершин выберем вершину  $u$  с минимальной меткой. Для соседних к  $u$  вершин, кроме отмеченных как посещенные, вычислим новую длину пути, равную сумме значений метки  $u$  и длины ребра, которое соединяет  $u$  с данным соседом.

3. В случае если рассчитанное на шаге 2 алгоритма значение меньше метки соседа, заменим значение метки на новое. Рассмотрев всех соседей, помечим вершину  $u$  как посещенную, необходимо повторить шаги алгоритма.

В результате применения алгоритма Дейкстры решения задачи о кратчайшем пути в графе, можем сформировать матрицу  $C^* = \{c_{ij}^*\}$ , где  $c_{ij}^*$  — время в пути от вершины  $i$  до вершины  $j$  на загруженной сети, при  $i, j \in \{1, \dots, |N|\}$ . Аналогом матрицы  $C^*$  на свободной от транспорта сети будет являться матрица  $C^0$ , элементы которой соответствуют времени перехода между вершинами графа  $G$  в случае отсутствия трафика.

## 2.2 Решение задачи распределения транспортных потоков на улично-дорожной сети г. Санкт-Петербурга

В ходе эксперимента была рассмотрена транспортная сеть города Санкт-Петербурга. Вершины графа  $G = (N, E)$ , соответствующие ей, обозначены точками на рисунке 1 в Приложении. Общее количество выбранных вершин 109, их соединяют 310 рёбер графа  $G$ . Вершины графы была выбраны на пересечении главных улиц и магистралей города, при этом вершина задавалась своим географическими координатами: долготой и широтой. На каждом ребре были заданы BPR-функции задержки с параметрами  $\alpha = 0,15$ ,  $\beta = 4$ . Время свободного движения  $t^0$  по ребру определялось как частное длины ребра и скорости, равной 40 км/ч.

Пропускная способность вычислялась по формуле  $p_i = k\mu p_i^0$ , где  $p_i^0$  — пропускная способность одной полосы,  $k$  — коэффициент многополосности,  $\mu$  — коэффициент светофорного регулирования [57]. Коэффициент многополосности принимался равным 1 в случае 1 полосы движения, 1.9 — в случае 2 полос, 2.7 — для 3 полос, 3.5 — для 4 и более полос. Коэффициент све-

тофорного регулирования показывает, как меняется пропускная способность перегона при задержке транспортных средств у светофора, и определяется по формуле

$$\mu = \frac{l}{l + \frac{v^2}{2} \left( \frac{1}{a} + \frac{1}{b} + v \frac{T_{all} - T_{green}}{2} \right)}, \quad (12)$$

$l$  — расстояние между перекрестками, м;  $a$  — ускорение при разгоне ( $1,0$  м/с<sup>2</sup>),  $b$  — замедление при торможении ( $1,5$  м/с<sup>2</sup>);  $all$  — продолжительность цикла регулирования светофора, с;  $t_{green}$  — продолжительность зеленой фазы светофора, с.

Пропускная способность одной полосы  $p_i^0$  вычисляется как

$$p_i^0 = \frac{3600v}{(v + 0,7 + 0,13v^2)}, \quad (13)$$

где  $v$  — средняя скорость движения по городу. Формула (11) учитывает средние значения следующих показателей: расстояние безопасности между остановившимися транспортными средствами, длина автомобиля, тормозной путь переднего автомобиля, коэффициент сопротивления качению, коэффициент сцепления, продольный уклон, отношение сцепного веса автомобиля к полному, скорость меньше 60 км/ч.

Период маршрутизации с 9 утра до 17 вечера был разбит на периоды  $Z_k = [z_{k-1}; z_k]$ , где  $k = 2, \dots, K$ . Совокупный транспортный спрос  $F^{rs}$  между районом отправления  $i$  и районом прибытия  $j$  задавался модельно для каждого из периодов  $Z_k$ .

Задача распределения транспортных потоков была решена алгоритмом Франка-Вульфа для каждого периода. С помощью алгоритма Дейкстры были сформированы матрицы переходов  $C^k$  для следующих периодов:

1. 9.00 – 11.00, матрица  $C^1$ ;
2. 11.00 – 12.00, матрица  $C^2$ ;
3. 12.00 – 14.00, матрица  $C^3$ ;
4. 14.00 – 15.00, матрица  $C^4$ ;
5. 15.00 – 17.00, матрица  $C^5$ .

### 3 Эксперимент

#### 3.1 Задача маршрутизации транспорта при переменных значениях потоков между районами отправления и прибытия

Во второй главе была решена задача распределения транспортных потоков и получены 5 матриц переходов, соответствующие пяти периодам дня. Для каждого ребра графа  $G$  транспортной сети г. Санкт-Петербурга зададим непрерывную кусочно-линейную функцию  $t_{ij}(b_i)$ , введённую в главе 1. Согласно лемме 1, функция вида (6) удовлетворяет свойству FIFO и определяет время перехода по ребру  $(i, j)$  в течение всего периода маршрутизации. Таким образом, она может быть использована в целевой функции при решении задачи TD-MDVRP.

Для решения задачи маршрутизации была сформирована матрица переходов  $C_{df}$  (*df*: *delay flow*) размера  $(109, 109)$ , каждый элемент которой является непрерывной кусочно-линейной функцией, задающей время перехода от узла  $i$  до  $j$  графа  $G$  в зависимости от момента отправления из узла  $i$ . Также была определена матрица  $C_{ff}$  (*ff*: *free flow*), соответствующая времени перехода на свободной от транспорта сети.

Тестовые примеры задачи TD-MDVRP имели размерности от 20 до 100 клиентов в каждой. Клиенты были выбраны случайным образом из множества вершин  $N$  графа  $G$ . Положим, что депо находятся во 2-ой и 90-ой узлах графа  $G$  (узел 2: пересечение КАД и Комендантского проспекта, узел 90: пересечение Санкт-Петербургского и Красносельского шоссе). В каждом из депо расположено по 10 машин одинаковой грузоподъёмности, равной 80 единиц. Распределение спроса (единиц товара) клиентов задачи было подчинено нормальному закону с параметрами  $N(20, 15)$ . В случае нарушения, накладыва-

емых на задачу ограничений, как правило, в целевую функцию задачи добавляют слагаемое, пропорциональное значению ошибки. Так как целью данной работы являлось сравнение времени прохождения построенных маршрутных планов задач TD-MDVRP и MDVRP, заданное для экспериментов количество машин в парке каждого депо было достаточным для генерации допустимого решения.

Тестовые задачи TD-MDVRP были решены генетическим алгоритмом. Количество поколений для генерации одного решения равнялось 20, при этом в популяции было 100 особей. Результаты представлены в таблице 1.

Количество клиентов	Количество машин			Общее время в пути, мин		
	Депо 1 (вершина 2)	Депо 2 (вершина 90)	Суммарно	Депо 1 (вершина 2)	Депо 2 (вершина 90)	Суммарно
20	1	3	4	34,14	391,31	425,45
30	1	4	5	34,14	493,69	527,83
40	2	4	6	170,84	502,12	672,96
50	2	6	8	169,11	698,24	867,35
60	6	6	12	533,87	737,26	1271,13
70	7	8	15	713,33	748,51	1461,84
80	4	8	12	385,6	919,48	1305,08
90	5	9	14	497,92	958,71	1456,63
100	8	9	17	874,69	1092,4	1967,09

**Таблица 1.** Решение задачи TD-MDVRP генетическим алгоритмом

Для каждой из 9 тестовых задач в таблице указано количество машин, необходимых для обслуживания заданного множества клиентов, а в последних 3 столбцах записаны значения целевой функции задачи TD-MDVRP, а именно суммарное время (мин.) в пути всех машин согласно сгенерированному маршрутному плану. Стоит отметить, что большая часть клиентов была назначена на обслуживание транспортным средствам из второго депо. Это

объясняется тем, что реализованный алгоритм назначал клиентов каждому из депо по признаку «близости» вершин графа задачи.

Также на тестовых задачи были проведены эксперименты при постоянной матрице переходов. То есть была решена задачи MDVRP, в которой матрица переходов была эквивалентна матрице расстояний между узлами графа  $G$ . Для произведения сравнения полученных результатов введём следующие обозначения.

Пусть  $s$  — решение тестового примера на свободной от транспорта сети,  $s^*$  — на загруженной. Введём функции  $len_{ff}(s)$  и  $len_{df}(s)$ , определяющие время прохождения маршрута  $s$  согласно матрицам  $C_{ff}$  и  $C_{df}$  соответственно. Сравнение двух постановок задач (TD-MDVRP и MDVRP) и соответствующих им решений приведено в таблице 2.

Количество клиентов	Количество машин	TD-MDVRP решение, мин.	MDVRP решение, мин		Среднее опоздание одной машины, мин.
			Ожидаемое	Реальное	
20	4	425,45	292,14	565,03	68,22
30	5	527,83	432,44	843,71	82,25
40	6	672,96	480,33	807,79	54,58
50	8	867,35	660,8	1241,51	72,59
60	12	1271,13	953,99	1615,52	55,13
70	15	1461,84	1089,92	1806,12	47,75
80	12	1305,08	986,07	1794,55	67,37
90	14	1456,63	1140,33	1730,71	42,17
100	17	1967,09	1372,19	2130,24	44,59

**Таблица 2.** Сравнение TD-MDVRP и MDVRP решений при переменных значениях потоков между районами отправления и прибытия

В столбце TD-MDVRP-решения таблицы 2 указаны значения  $len_{df}(s^*)$ , полученные ранее. Графа MDVRP-решения представлена двумя показателями: ожидаемое и реальное время. Ожидаемое время означает планируемое

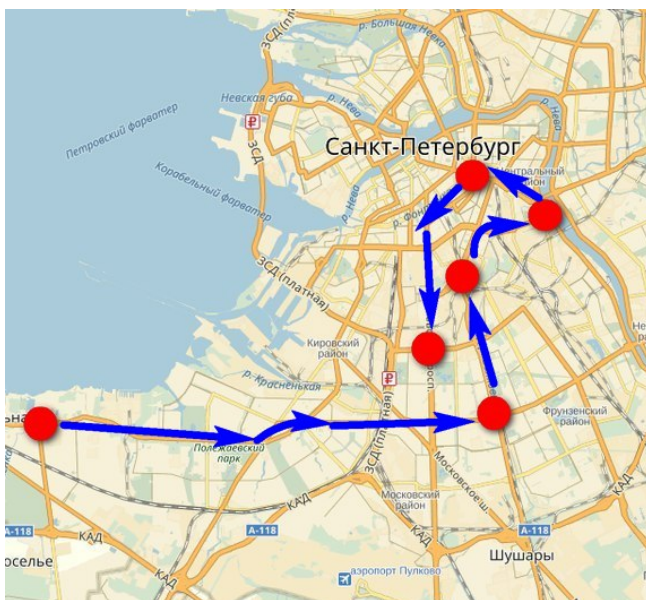


время  $len_{ff}(s)$ , которое затратят транспортные средства согласно MDVRP-решению. Так как постановка задачи MDVRP не учитывает фактор трафика при маршрутизации, реальное время в пути будет отличаться от запланированного. Для получения значения реального времени наложим маршрутный план  $s$  на матрицу переходов  $C_{df}$ , представляющую время перехода по загруженной сети. Получим столбец значений  $C_{df}(s)$ , обозначенный в таблице 2 как «Реальное время» в пути MDVPR-решения.

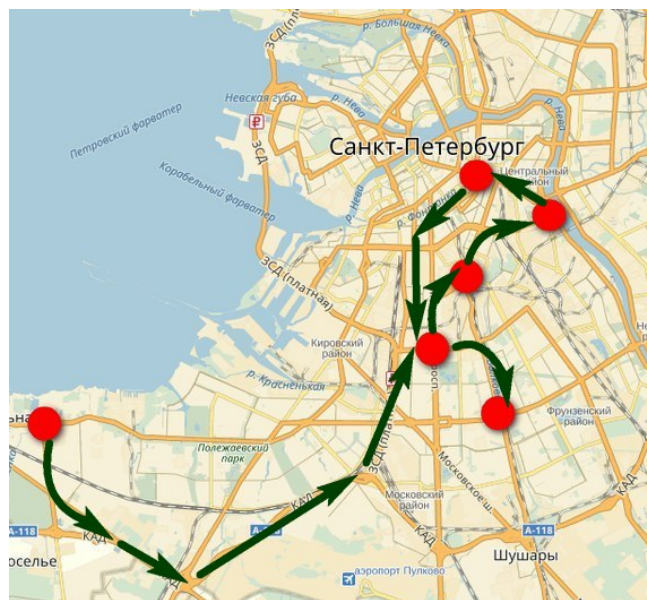
В итоге имеем, что во всех рассмотренных тестовых задачах реальное время, необходимое для посещения всех клиентов MDVRP-решения, оказалось больше, чем данный показатель у TD-MDVRP-решения. Стоит отметить, что такой же результат наблюдался при рассмотрении задачи маршрутизации TSP с постоянных значениях потоков между районами отправления и прибытия [61]. В последнем столбце таблицы 2 показано значение средней задержки одного транспортного средства, т.е. на сколько в среднем увеличиться продолжительность маршрута в действительности по сравнению с запланированным временем.

Детально рассмотрим маршрутный план машины, следующей из депо в 90-ом узле графа и сравним VRP- и TD-VRP-решения задачи с пятью клиентами: 60, 62, 77, 85, 102. Примем время обслуживания одного клиента равным 10 минутам.

Согласно генетическому алгоритму реализация VRP-решения с возвратом в депо заняла 253 минуты, TDVRP-решения — 232 минуты. Маршруты движения представлены на рисунках 6 и 7 соответственно. TD-решение строит маршрут объезда клиента, не включая загруженные в часы пик участки Санкт-Петербургского и Петергофского шоссе, выбирая вместо них объезд-



**Рис. 6.** VRP-решение задачи  
с клиентами в узлах 60, 62, 77, 85, 102



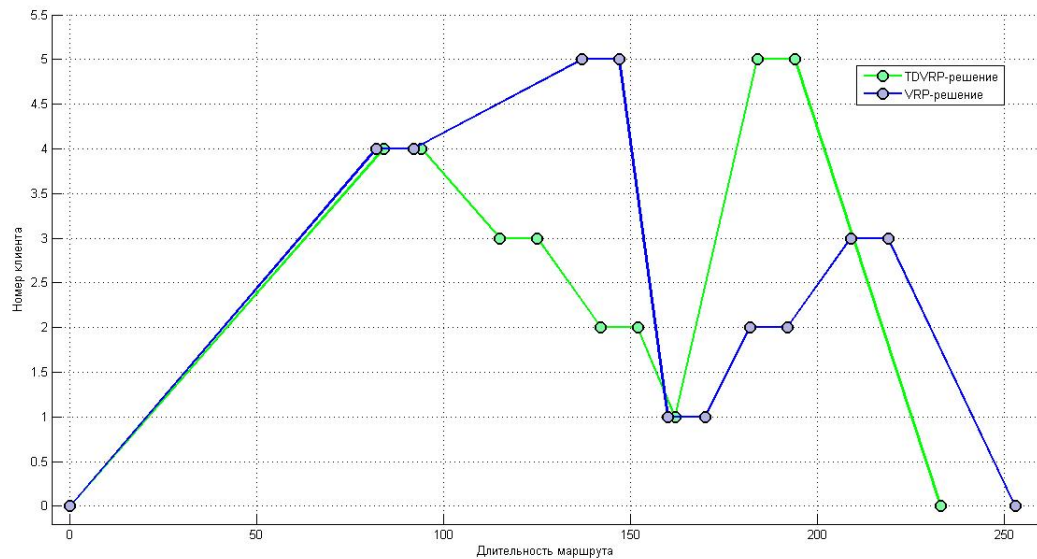
**Рис. 7.** TDVRP-решение задачи  
с клиентами в узлах 60, 62, 77, 85, 102

ные дороги (КАД, ЗСД). На рисунке 8 показана динамика изменения времени в пути VRP и TD-VRP решений.

### 3.2 Задача маршрутизации транспорта при переменных значениях времени перехода между узлами сети

В ходе эксперимента была рассмотрена транспортная сеть города Санкт-Петербурга. Граф сети содержит 109 вершин и 255 рёбер. Значения времени перехода между узлами графа были получены через сервис Яндекс Карты с помощью API Яндекс.Карт. API Яндекс.Карт представляет собой набор компонентов для размещения интерактивных Яндекс.Карт на страницах сайта или в веб-приложении. С помощью API можно отображать карты с различными объектами, искать адреса, прокладывать маршруты, строить свои схемы и многое другое. Интерфейс требует поддержки JavaScript [62].

Каждый узел рассматриваемого графа был задан с помощью двух ко-



**Рис. 8.** Динамика изменения времени в пути VRP- и TDVRP-решения задачи с 5 клиентами

ординат: долготы и широты. Реализованный в классе router API Яндекс.Карт метод `getHumanJamsTime()` возвращает строковое представление времени проезда маршрута с единицами измерения с учетом пробок. Для этого необходимо передать в функцию `router` массив точек, а именно 4 координаты, задающие одно ребро графа. Результат вычисления маршрута поступает в функцию-обработчик в виде коллекции `GeoObjectCollection`. Этот объект реализует интерфейс `IGeoObject`, описывающий проложенный маршрут. Метод `getHumanJamsTime()` позволяет получить время перехода между 2 заданными точками на графе.

Эксперимент проводился с 15 февраля по 1 марта, каждый час с 08:00 до 20:00. По окончании эксперимента для каждого ребра было рассчитано среднее значение времени перехода по нему для каждого из 13 периодов. С помощью алгоритма Дейкстры поиска кратчайшего пути на графе сформирована матрица переходов, каждый элемент которой имеет вид (6). Матрица переходов, соответствующая свободной от трафика сети, была получена по

данным о времени перехода между узлами сети в ночное время суток.

Как и в предыдущем эксперименте, тестовые примеры задачи TD-MDVRP имели размерности от 20 до 100 клиентов в каждой. Клиенты были выбраны случайным образом из множества вершин  $N$  графа  $G$ . Депо находилось во 2-ой и 90-ой узлах графа  $G$  (узел 2: пересечение КАД и Комендантского проспекта, узел 90: пересечение Санкт-Петербургского и Красносельского шоссе). В каждом из депо расположено по 5 машин одинаковой грузоподъёмности, равной 300 единиц. Распределение спроса (единиц товара) клиентов задачи было подчинено нормальному закону с параметрами  $N(20, 15)$ . Частный случай задачи MDVRP, задача маршрутизации VRP с одним депо и несколькими транспортными средствами, был представлен в работе [63].

Ниже в таблице 3 приведено сравнение трёх величин: суммарного времени на реализацию TD-MDVRP-решения —  $len_{df}(s^*)$ , ожидаемого времени реализации MDVRP-решения —  $len_{ff}(s)$  и реального времени в пути согласно MDVRP-решению —  $len_{df}(s)$ . Также приведены значения среднего опоздания одной машины, т.е. выхода за лимит времени, запланированного перед началом движения.

Так же как и в задаче маршрутизации при переменных значениях потоков между районами отправления и прибытия, использование TD-модели при построении маршрутных планов по матрице переходов, сформированной по данным от Яндекса, позволяет генерировать решения с меньшим временем в пути, чем классическая постановка задачи MDVRP.

Рассмотрим применение алгоритма динамической адаптации, описанного в главе 1 и работах [43, 50] и сравним его с генетической эвристикой. Процедура динамической адаптации применялась после назначения обслуживания

Количество клиентов	Количество машин	TD-MDVRP решение, мин.	MDVRP решение, мин		Среднее опоздание одной машины, мин.
			Ожидаемое	Реальное	
20	2	427,83	312	465,41	76,7
30	2	500,58	342	526,05	92,02
40	3	609,7	451	649,65	66,21
50	3	706,4	530	786,33	85,44
60	4	780,27	580	862,36	70,59
70	6	981,23	696	1049,28	58,88
80	6	1023,73	737	1115,06	63,01
90	6	1121,78	880	1309,51	71,58
100	7	1233,56	890	1322,81	61,83

**Таблица 3.** Сравнение TD-MDVRP и MDVRP решений при переменных значениях времени перехода между узлами сети

клиентов одному из депо, для маршрута каждого транспортного средства в отдельности. Количество запусков, для генерации решения на одном периоде, равнялось пяти. Параметры алгоритма ДУГА:  $T = 5$ ,  $N = 5$ .

Статистику изменений значения целевой функции проведём по формуле

$$k(p) = \frac{l_{GA}(p) - l_{DAGA}(p)}{l_{GA}(p)}. \quad (14)$$

В формуле (15) через  $l_{GA}(p)$  обозначено суммарное время в пути согласно маршрутному плану, сгенерированному генетической эвристикой для тестового примера  $p$  и  $l_{DAGA}(p)$  — с помощью процедуры динамической адаптации генетического алгоритма. Результаты предложенной оценки показаны в таблице 4.

Для всех тестовых задач, приведённых в таблице 4, значение общего времени в пути согласно маршрутному плану, полученному алгоритмом ДУГА, меньше, чем генетическим алгоритмом. Можно заметить, что предложенный новый поход требует дополнительных запусков эвристики на каждом пери-

Количество клиентов	Количество машин	Генетический алгоритм (общее время), мин.	Динамическая адаптация (общее время), мин.	k, 100 %
20	2	427,83	419,27	2
30	2	500,58	479,56	4,2
40	3	609,7	584,70	4,1
50	3	706,40	681,68	3,5
60	4	780,27	759,91	2,6
70	6	981,23	950,78	3,1
80	6	1023,73	980,7	4,2
90	6	1121,78	1057,77	5,7
100	7	1233,56	1157,08	6,2

**Таблица 4.** Сравнение алгоритма ГА и ДУГА решения задачи MDVRP

оде, тем самым увеличивая время получения одного маршрутного плана по сравнению с классическим алгоритмом. Однако при фиксированном одинаковом времени работы метод динамической адаптации генетической эвристики способен генерировать лучшие решения по сравнению с генетическим алгоритмом. Данное замечание стоит учитывать в задачах большой размерности, когда время генерации одного маршрута может занимать десятки минут. В случае заданного промежутка времени на получение маршрутного плана, выгоднее использовать процедуру динамической адаптации вместо классической эвристики, распределив время между периодами работы алгоритма.

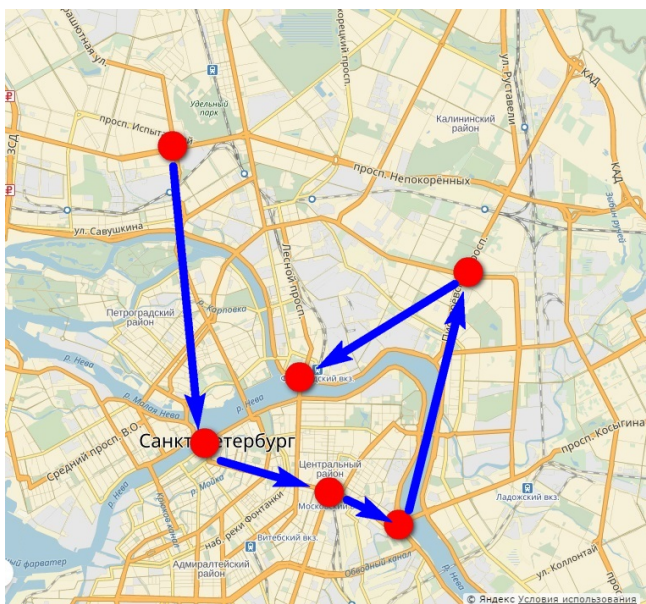
В случае, если одной из задач компании, осуществляющую доставку товаров клиентам, является минимизация общего времени в пути транспортных средств, имеет смысл заранее выбирать период дня, начиная с которого осуществлять доставку. Сравнение зависимости суммарного времени в пути машин от времени начала маршрутизации представлено в таблице 5.

В таблице 5 наблюдается ярко выраженный вечерний час пик — 18:00,

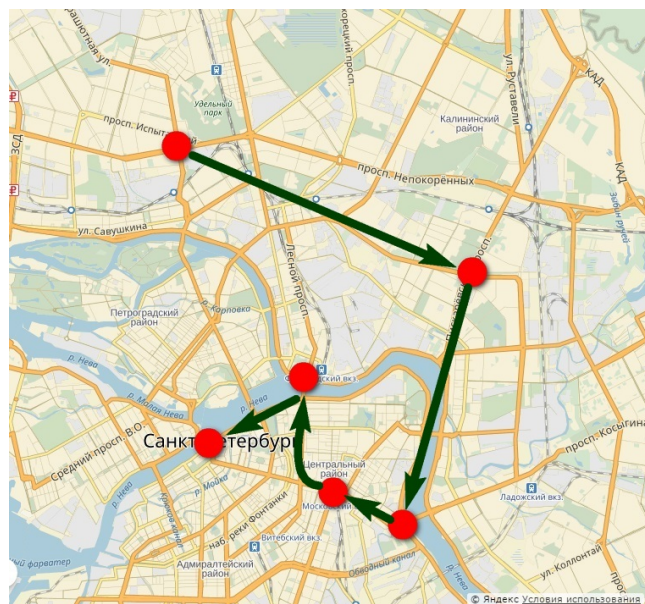
Количество клиентов	Время отправления из депо						
	8:00	10:00	12:00	14:00	16:00	18:00	20:00
	TD-MDVRP решение, мин.						
20	427,83	458,32	462,83	471,31	442,82	485,32	392,43
30	500,58	515,43	513,35	523,13	514,14	508,07	477,62
40	609,70	647,03	621,25	656,13	611,05	672,78	628,24
50	706,40	764,33	740,23	791,95	712,57	815,47	696,76
60	780,20	770,32	864,78	870,45	901,54	1042,23	869,34
70	981,20	1020,43	973,55	1050,79	1009,58	1087,62	1037,90
80	1023,70	1026,20	1108,13	1196,95	1158,30	1119,37	1032,52
90	1121,70	1337,91	1290,87	1545,15	1358,60	1484,15	1317,77
100	1233,56	1277,79	1325,98	1404,72	1473,43	1341,50	1312,65

**Таблица 5.** Зависимость общего времени в пути от времени начала маршрутизации в утренние часы и днём (с 8:00 до 14:00) время движения по дорогам отличается мало. Наиболее выгодным временем отправления транспортных средств для рассмотренных тестовых задач оказалось время 16:00. Стоит отметить, что время движения в пути зависит от топологии расположения клиентов. Таким образом, для разных вариантов задач оптимальное время осуществления маршрутизации может отличаться. Данное значение необходимо выбирать для каждой задачи в отдельности по накопленной статистике.

Рассмотрим пример маршрута движения одного транспортного средства, которому необходимо обслужить клиентов 12, 27, 34, 48, 59, 61 на графе г. Санкт-Петербурга. Построенные маршрутные планы представлены на рисунках 8 и 9. Время начала маршрутизации полагалось равным 8 утра. VRP-модель задачи сгенерировала маршрут в 88 минут, включающий в себя загруженные в часы пик участки Петроградской стороны и весь Невский проспект. В то время как, TD-VRP решение строит маршрут в объезд основных заторов, и занимает меньше времени, а именно 76 минут.



**Рис. 9.** VRP-решение задачи  
с клиентами в узлах 12, 27, 34, 48, 59, 61



**Рис. 10.** TDVRP-решение задачи  
с клиентами в узлах 12, 27, 34, 48, 59, 61



## 4 Заключение

В данной работе была исследована задача маршрутизации транспорта с несколькими депо (MDVRP) и задача маршрутизации транспорта с несколькими депо при переменных значениях времени перехода между узлами (TD-MDVRP). Был дан подробный обзор математических моделей задачи маршрутизации транспорта на сети мегаполиса. Описывается опыт применения данной концепции на улично-дорожных сетях городов Германии, Италии, Великобритании, США, Китая.

Была проведена классификация существующих эвристических методов решения задач маршрутизации. Для вычислительных экспериментов был выбран и реализован на языке Java генетический алгоритм. Матрицы переходов, соответствующие разным периодам дня, были получены двумя способами.

В первом из них алгоритмом Франка–Вульфа была решена задача нахождения равновесного по Вардропу распределения транспортных потоков города с линейной BPR-функции задержки. Исходные данные в рассматриваемой задаче задавались модельно. Для получения действительного распределения транспортных потоков необходимо точно знать существующие матрицы корреспонденций транспорта. Второй способ использовал информацию о трафике, полученную через сервис Яндекс.Пробки. Накопленная статистика о загруженности улично-дорожной сети содержала 2 недели наблюдений. Матрицы переходов в обоих случаях были сформированы алгоритмом Дейкстры поиска кратчайшего пути на графе.

Для проведения вычислительных экспериментов была рассмотрена транспортная сеть г. Санкт-Петербурга, состоящая из 109 узлов. Выбранные узлы находились на пересечении главных улиц и магистралей города. Сгене-

нированные случайным образом тестовые задачи для маршрутизации включали от 20 до 100 клиентов. Было проведено сравнение результатов маршрутизации согласно постановкам задач MDVRP и TD-MDVRP. Как итог, решения задачи MDVRP, полученные без учёта информации о загруженности сети, существенно уступают во времени маршрутам TD-MDVRP, использующим её при генерации. Также наблюдается выход маршрутов движения за горизонт планирования, т.е. имеет место опоздание транспортных средств по сравнению со временем, запланированным перед началом движения.

Таким образом, фактор трафика в условиях загруженной улично-дорожной сети мегаполиса играет существенную роль и не должен быть проигнорирован в рамках построения маршрутных планов. При том же объёме используемых ресурсов выбор правильной математической модели при решении задач маршрутизации позволит увеличить прибыль компании.

## Литература

1. Лукинский В.С. и др. Модели и методы теории логистики. Учебное пособие. Спб.: Питер. 2003. 219 с.
2. Захаров В. В. Методы и модели прикладной математической логистики // Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 742–776.
3. Захаров В. В., Крылатов А. Ю. Конкурентная маршрутизация транспортных потоков поставщиками услуг навигации // Управление большими системами. 2014. № 49. С. 129–147.
4. Sarikli, D., Powell, S. A heuristic method for the open vehicle routing problem // Journal of the Operation Research Society. 2000. Vol. 51 (5). P. 564 – 573.
5. Crevier, B., Cordeau, J.F., Laporte, G. The multi-depot vehicle routing problem with inter-depot routes // European Journal of Operational Research. 2007. Vol. 176 (2) P. 756–773.
6. Hoff, A., Gribkovskaia I., Laporte, G., Lokketangen, A. Lasso solution strategies for the vehicle routing problem with pickups and deliveries // European Journal of Operational Research. 2007. Vol. 192 (3). P. 755–766.
7. Toth, P., Vigo, D. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia. 2002. 367 p.
8. Francis, P., Smilowitz, K. Modeling techniques for periodic vehicle routing problems // Transportation Research B. 2006. Vol. 40 (10). P. 872–884.
9. Gendreau, M., Laporte G., Seguin, R. An exact algorithm for the vehicle routing problem with stochastic demands and customers // Transportation Science. 1995. Vol. 29. P. 143–155.

10. Dror, M., Trudeau, P. Savings by split delivery routing // *Transportation Science*. 1989. Vol. 23. P. 141–145
11. Bard, J.F., Huang, L., Dror, M., and Jaillet, P. A Branch and Cut Algorithm for the VRP with Satellite Facilities // *IIE Transactions* Vol. 30. P. 821-834
12. Pillac, V., Gendreau, M., Gueret, C., Medaglia, A.L. A review of dynamic vehicle routing problems // *European Journal of Operational Research*. 2013. Vol. 225 (1). P. 1–11.
13. Malandraki, C. and Daskin, M. S. Time dependent vehicle routing problems: formulations, properties and heuristic algorithms // *Transportation Science*. 1992. Vol. 26. P. 185–200.
14. Malandraki, C., Dial, R.B. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem // *European Journal of Operational Research*. 1996. Vol. 90. P 45–55.
15. Hill, A. V. and Benton, W. C. Modelling Intra-City Time-Dependent Travel Speeds for Vehicle Scheduling Problems // *Journal of the Operational Research Society*. 1992. Vol. 43. P. 343–351.
16. Ichoua, S. and Gendreau, M. and Potvin, J-Y. Vehicle dispatching with time-dependent travel times // *European Journal of Operational Research*. 2003. Vol. 144 (2). P 379–396.
17. Mancini, S. Time Dependent Travel Speed Vehicle Routing and Scheduling on a Real Road Network: The Case of Torino // *Transportation Research Procedia*. 2014. Vol. 3. P 433–441.
18. Fleischmann, B., Gietz, M., Gnutzmann, S.. Time-Varying Travel Times in Vehicle Routing // *Transportation Science*. 2004. Vol. 38 (2). P. 160–173.

19. Donati, A.V., Montemanni R., Casagrande N., Rizzoli, A.E., Gambardella, L.M. Time dependent vehicle routing problem with a multi ant colony system // European Journal of Operational Research. 2008. Vol. 185 (3). P. 1174–1191.
20. Kuo Y., Wang C.-C., and Chuang P.-Y. Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds // Computers Industrial Engineering. 2009. Vol. 57 (4). P. 1385–1392.
21. Eglese, R., Maden, W., and Slater, A. A road timetableTM to aid vehicle routing and scheduling // Computers and Operations Research. 2006. Vol. 33 (12). P. 3508–3519.
22. Maden, W., Eglese, R., and Black, D. Vehicle routing and scheduling with time-varying data: A case study // Journal of the Operational Research Society. 2010. Vol. 61 (3). P. 515–522.
23. Franceschetti, A., Honhon, D., Van Woensel, T., Bektas, T., and Laporte, G. The time-dependent pollution-routing problem. Transportation // Research Part B: Methodological. 2013. Vol. 56. P. 265–293.
24. Zhang, T., Chaovalitwongse, W.A., and Zhang, Y. Integrated Ant Colony and Tabu Search approach for time dependent vehicle routing problems with simultaneous pickup and delivery // Journal of Combinatorial Optimization. 2004. Vol. 28 (1). P. 288–309.
25. Cho, D. W., Lee, Y. H., Lee, T. Y., and Gen, M. An adaptive genetic algorithm for the time dependent inventory routing problem // Journal of Intelligent Manufacturing. 2014. Vol. 25 (5). P. 1025–1042.

26. Gendreau, M., Ghiani, G., and Guerriero, E. Time-dependent routing problems: A review // Computers and Operations Research. 2015. Vol. 64. P. 189–197.
27. Solomon, M.M. Algorithms for the vehicle routing and scheduling problems with time window constraints. 1987 // Operations Research. Vol. 35. P. 254–265.
28. Figliozzi, M.A. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics // Transportation Research Part E: Logistics and Transportation Review. 2012. Vol. 48 (3). P. 616–636.
29. Kumar, S. and Panneerselvam, R. A. Time-Dependent Vehicle Routing Problem with Time Windows for E-Commerce Supplier Site Pickups Using Genetic Algorithm // Intelligent Information Management. 2015. Vol. 7. P. 181–194.
30. Talbi, El-Gh. Metaheuristics for Bi-level Optimization. Springer Publishing Company, Incorporated, 2013. 306 p.
31. Giosa, I. D., Tansini, I. L., and Viera, I. O. New assignment algorithms for the multi-depot vehicle routing problem // Journal of the Operational Research Society. 2002. Vol. 53. P. 977–984.
32. Wu, T.-H., Low, Ch., Bai, J.-W. Heuristic solutions to multi-depot location-routing problems // Computers and Operations Research. 2002. Vol. 29. P. 1393–1415.

33. Creviera, B., Cordeau, J.-F., and Laporte, G. The multi-depot vehicle routing problem with inter-depot routes // *European Journal of Operational Research*. 2007. Vol. 176. P. 756–773.
34. Zachariasen, M. and Dam, M. Tabu Search on the Geometric Traveling Salesman Problem. In I. H. Osman and J. P. Kelly, editors, *Metaheuristics: theory and applications* // *In Proceedings from Metaheuristics International Conference, Colorado*. 1996. P. 571–587.
35. Surekha, P., Sumathi, S. Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms // *World Applied Programming*. 2011. Vol. 1 (3). P. 118–131.
36. NEO Web: Networking and Emerging Optimization. University of Malaga, Spain.  
<http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/> (дата обращения: 28.03.2017)
37. Holland, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI. 1975. 183 p.
38. Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., and Dizdarevic, S. Genetic algorithms for the travelling salesman problem: A review of representations and operators // *Artificial Intelligence Review*. 1999. Vol. 13(2). P. 129–170.
39. Liu, Ch. and Kroll, A. On designing genetic algorithms for solving small- and medium-scale traveling salesman problems // *In International Symposium on Swarm Intelligence and Differential Evolution, SIDE*. 2012. P. 283–291.

40. Goldberg, D. E. and Lingle, R. Alleles, loci, and the traveling salesman problem // In Proc. of the International Conference on Genetic Algorithms and Their Applications. 1985. P.154–159.
41. Oliver, I. M., Smith, D. J., and Holland, J. R. C. A study of permutation crossover operators on the traveling salesman problem // In Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application. 1987. P. 224–230.
42. Puljic, K., Manger, C.R. Comparison of eight evolutionary crossover operators for the vehicle routing problem // Mathematical Communications. 2013. Vol. 18. P. 359-375.
43. Мугайских А. В. Динамическая адаптация генетического алгоритма для задачи коммивояжёра // Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 665–670.
44. Viacheslav A. Shirokikh, Victor V. Zakharov Dynamic Adaptive Large Neighbourhood Search for Inventory Routing Problem // Advances in Intelligent Systems and Computing, 2015. Vol. 359. P. 231–241
45. V. V. Zakharov and A. N. Shchegryaev Stable Cooperation in Dynamic Vehicle Routing Problems // Automation and Remote Control. 2015. Vol. 76 (5). P. 935–943
46. Bellman, R. Dynamic Programming. Princeton University Press, Princeton, NJ, USA. 1957. 384 p.
47. Мугайских А.В. Динамическая адаптация генетического алгоритма для решения задач транспортной маршрутизации. Выпускная квалификацион-



- ная работа бакалавра, Санкт-Петербургский государственный университет. 2015. 50 с.
48. Reinelt, G. Travelling Salesman Problem Library, 2008.  
<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/> (дата обращения: 28.03.2017)
49. Борисовский П.А., Еремеев А.В. О сравнении некоторых эволюционных алгоритмов // Автоматика и телемахинка. 2004. Т. 3. С. 3–9.
50. Захаров В.В., Мугайских А.В. Динамическая адаптация эвристических алгоритмов для задач маршрутизации транспорта. // В сборнике: Логистика: современные тенденции развития. Материалы XV Международной научно-практической конференции. 2016. С. 131-134.
51. Петросян Л.А., Зенкевич Н. А. Принципы устойчивой кооперации // Управление большими системами: сборник трудов. 2009. Т.3. С. 100–120.
52. Zakharov V.V., Dementieva M. Multistage cooperative games and problem of time consistency // International Game Theory Review, 2004. Vol. 6. P. 157–170.
53. Zakharov V.V., Shchegryaev A.N. Multi-period cooperative vehicle routing games // Contributions to Game Theory and Management, 2014. Vol. 7(2). P. 349–359.
54. Wardrop, J. G. Some theoretical aspects of road traffic research // Proc. Institution of Civil Engineers. 1952. Vol. 2. P. 325–378.
55. Patriksson, M. The Traffic Assignment Problem: Models and Methods. 2015. Dover Books on Mathematics. 240 p.

56. Dijkstra, E. W. A note on two problems in connection with graphs. *Numerische Mathematik*. 1959. Vol. 1. P. 269–271.
57. Булавина Л. В. Расчет пропускной способности магистралей и узлов. Екатеринбург: ГОУ ВПО УГТУ–УПИ, 2009. 44 с.
58. Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J. Y., and Semet F. A guide to vehicle routing heuristics // *Journal of the Operational Research Society*. 2002. Vol. 53 (5). P. 515-522.
59. Cordeau, J.-F., Gendreau, M., and Laporte, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems // *Networks*. 1997. Vol. 30 (2). P. 105–119.
60. Ausiello, G., Escoffier, B., Monnot, J., Paschos, V. Reoptimization of minimum and maximum traveling salesman's tours // *Journal. of Discrete Algorithms*. 2009. Vol. 7 (4). P. 453–463.
61. Мугайских А.В. О задаче коммивояжёра на сети мегаполиса в условиях распределения транспортных потоков по Вардропу // *Процессы управления и устойчивость*. 2016. Т. 3. № 1. С. 680-684.
62. Яндекс Карты API  
<https://tech.yandex.ru/maps/jsapi/> (дата обращения: 28.03.2017)
63. Захаров В.В., Мугайских А.В. О задаче маршрутизации трафика с учётом трафика // *Транспорт России: проблемы и перспективы – 2016*. 2016. Т.1. С. 218–222.

## Приложение

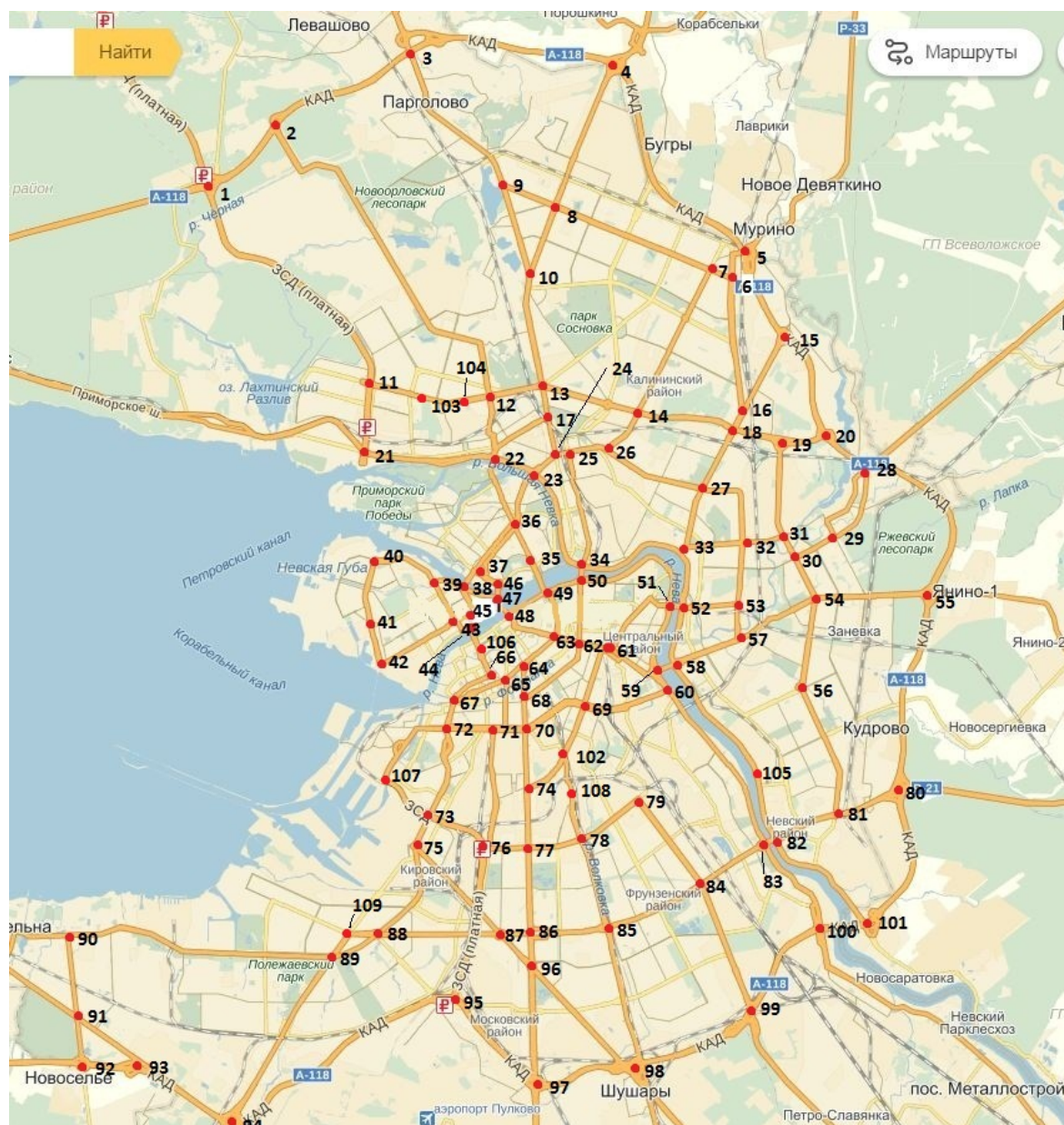


Рис. 1. Улично-дорожная сеть г. Санкт-Петербурга, 109 узлов